Improving Sample Quality of Diffusion Models Using Self-Attention Guidance Supplemental Material

Susung Hong Gyuseong Lee Wooseok Jang Seungryong Kim
Korea University, Seoul, Korea
{susung1999, jpl358, jws1997, seungryong_kim}@korea.ac.kr

In this document, we provide additional details of DDPM [3], implementation details of our method, more analyses and results, and the human evaluation protocol. We also discuss the limitations and future work at the end.

A. Denoising Diffusion Probabilistic Models

DDPM [6] is a generative model that generates an image from white noise with iterative denoising steps. Given an image $\mathbf{x}_0$ and a variance schedule β_t for an arbitrary timestep $t \in \{1, 2, \dots, T\}$, the forward process of DDPM is defined as a Markov process of the form:

$$q(\mathbf{x}_{t+1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+1}; \sqrt{1-\beta_t}\mathbf{x}_t, \beta_t\mathbf{I}). \tag{1}$$

Note that we can directly get $\mathbf{x}_t$ from $\mathbf{x}_0$ in the closed form:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2}$$

where $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Similarly, the reverse process is defined as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)\mathbf{I}), \tag{3}$$

where μ_θ and Σ_θ denote neural networks with parameter θ.

For the training phase, with Σ_θ fixed to a constant $\sigma_t^2 = \beta_t$ as in DDPM, $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is compared with the following forward posterior:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_0, \mathbf{x}_t), \tilde{\beta}_t\mathbf{I}), \tag{4}$$

where $\tilde{\mu}_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t$, and $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. However, instead of directly comparing μ_θ to $\tilde{\mu}_t$, Ho *et al.* [6] discover that it is beneficial to optimize ϵ_θ with the following simplified objective after reparameterization:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{5}$$

$$L_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0,t,\boldsymbol{\epsilon}}[||\boldsymbol{\epsilon} - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)||^2]. \tag{6}$$

For sampling $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, we can compute the following from $\mathbf{x}_T$ to $\mathbf{x}_0$:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)) + \sigma_t\mathbf{z}, \tag{7}$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. Rewriting Eq. 5, we can get $\hat{\mathbf{x}}_0$ which is a prediction of $\mathbf{x}_0$ at each timestep with the following formula:

$$\hat{\mathbf{x}}_0 = (\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t, t))/\sqrt{\bar{\alpha}_t}. \tag{8}$$

B. Additional Implementation Details

B.1. Environmental setting

For the experiments, we use two servers of 8 NVIDIA GeForce RTX 3090 GPUs each to sample from the pre-trained models of ADM [3], IDDPM [9], Stable Diffusion v1.4 [12], and DiT [11]. We build upon the PyTorch [10] implementation of these models, taking all the weights for our experiments from their publicly available repository.

B.2. Selective blurring

In practice, we efficiently implement selective blurring in Sec. 5.2. At the first step, we blur the intermediate reconstruction $\hat{\mathbf{x}}_0$ of $\mathbf{x}_t$ [6]. Then, we apply masks $1 - M_t$ and M_t on $\hat{\mathbf{x}}_0$ and the blurred version of $\hat{\mathbf{x}}_0$, respectively. Finally, we aggregate the output and then noise it again with the predicted noise $\epsilon_\theta(\mathbf{x}_t)$ that we use for computing $\hat{\mathbf{x}}_0$ above. This process ends up producing the same $\widehat{\mathbf{x}}_t$ as Eq. 15 in the main paper.

B.3. Combination of SAG and CFG

Naïvely, in order to combine SAG with CFG [7] in Stable Diffusion [12] and DiT [11], we have to compute SAG through the conditional and unconditional models, which requires us four feedforward steps. In practice, the guided prediction of noise can be efficiently calculated as follows:

$$\tilde{\epsilon}(\mathbf{x}_t) = \epsilon_\theta(\mathbf{x}_t, c) + s_{\mathrm{c}}(\epsilon_\theta(\mathbf{x}_t, c) - \epsilon_\theta(\mathbf{x}_t)) + s_{\mathrm{s}}(\epsilon_\theta(\mathbf{x}_t) - \epsilon_\theta(\bar{\mathbf{x}}_t)), \tag{9}$$

where s_{c} and s_{s} denote the scales of CFG and SAG, respectively, and c denotes a text prompt.

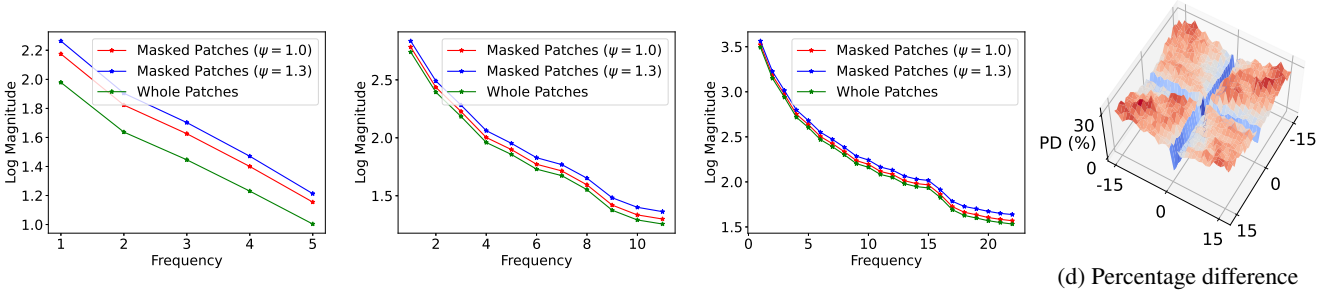
B.4. Hyperparameter settings

In Table 1, we report our hyperparameter settings for our experiments. In the ablation studies in the main paper, we set the other parameters to the constants in Table 1, while testing the ablated parameter. Note that σ is dependent on the input resolution.

Model		Self-attention parameter			Gaussian-blur parameter
		Guidance scale	Threshold	Layer	σ
ADM [3]	ImageNet 256×256 (unconditional)	0.5, 0.8	1.0	Output 2	9
	ImageNet 256×256 (conditional)	0.2	1.0	Output 2	9
	LSUN Cat 256×256	0.05	1.0	Output 2	9
	LSUN Horse 256×256	0.01	1.0	Output 2	9
	ImageNet 128×128	0.1	1.0	Output 8	3
IDDPM [9]	ImageNet 64×64 (unconditional)	0.05	1.0	Output 7	1
Stable Diffusion [12]		0.75, 1.0	1.0	Middle	1
DiT [11]		0.005	1.0	13th block	1

Table 1: Hyperparameter settings.

Figure 1: **Comparison between self-attention masks of DINO [1] and ADM [3]:** (a) the self-attention masks extracted from DINO [1], (b) the self-attention masks extracted from ADM [3].



(a) Frequency-magnitude plot of 8×8 patches

(b) Frequency-magnitude plot of 16×16 patches

(c) Frequency-magnitude plot of 32×32 patches

(d) Percentage difference (PD) in frequency domain

Figure 2: **Frequency analysis of the self-attention masks:** (a), (b) and (c) show the frequency-magnitude graphs of 8×8, 16×16, and 32×32 patches, respectively. ψ denotes the masking threshold. (d) is a 3D visualization that shows the percentage difference of magnitude between masked and non-masked patches in the frequency domain regarding the 32×32 patches.

C. Additional Analyses and Results

C.1. Exploring the self-attention in diffusion models

We show the visualizations of self-attention maps in the 8×8, 16×16, and 32×32 resolutions of the U-Net [13] of ADM [3] in Fig. 5. The attention maps at $t = 0, 49, 99, 149, 199, 249$ are visualized at each row in order, and the layers are aligned left to right. In this visualization, can see that the attention maps at the intermediate timesteps capture the structure of generated images. Also, we extract the self-attention masks from the different heads and layers from the U-Net and visualize them in Fig. 6 and Fig. 7. *Average* in this figure means the obtained masks after averaging attention maps of the four heads. Moreover, we compare the self-attention masks of ADM with those of DINO [1] in Fig. 1. Compared to the attention masks of DINO, those of ADM are more attending to multiple objects and high-frequency details of the generated images where diffusion models have to elaborate.

Based on the observation, we are interested in two aspects that the self-attention of diffusion models attends to: the frequency and the semantics of the samples. Therefore, we first investigate how the self-attention maps correlate with frequency by comparing the frequency spectra of patches with high attention scores to those of all patches. We observe that high-attention patches contain more high-frequency details (Fig. 2). We then evaluate how the self-attention maps align with foreground objects (Table 2 and Fig. 3) and discover that they capture some semantic information at all resolutions.

C.2. Additional ablation studies

We conduct experiments on the threshold of self-attention masking that affects the ratio of the blurred region with 10k samples. We test the thresholds of 0.7, 1.0, and 1.3. As shown in Table 3, the highest metrics are obtained when the threshold

Figure 3: **Visualization of self-attention masks compared to object masks.** Generated images (top row), the object masks of Mask R-CNN [5] (middle row), and the self-attention masks of unconditional ADM [3] (bottom row).

Patch size	ψ	Random	Self-attn.	% Diff.
8×8	1.0	0.16	0.23	+ 44%
	1.3	0.09	0.14	+ 56%
16×16	1.0	0.18	0.25	+ 39%
	1.3	0.05	0.11	+ 120%
32×32	1.0	0.18	0.26	+ 44%
	1.3	0.04	0.10	+ 150%

Table 2: **Semantic analysis of the self-attention masks.** ψ denotes the masking threshold, and % Diff. denotes the percentage difference of the IoU over the random counterpart.

ψ	Baseline	$\psi = 0.7$	$\psi = 1.0$	$\psi = 1.3$
FID (↓)	5.98	5.67	**5.47**	5.66
IS (↑)	141.72	148.60	**151.12**	145.58

Table 3: **Ablation study of the masking threshold (ψ).** The results are derived from ADM trained on ImageNet 128×128.

Layer	Baseline	In. 11	In. 8	Mid.	Out. 2	Out. 5	Out. 8
FID (↓)	5.98	5.54	5.61	5.63	5.59	5.57	**5.47**
IS (↑)	141.72	150.07	148.20	143.44	150.62	141.73	**151.12**

Table 4: **Ablation study of the layer where we extract the attention map.** The results are derived from ADM trained on ImageNet 128×128. We denote the middle block as *Mid.*, and the nth layer of the input and output blocks as *In.* n and *Out.* n, respectively.

value is 1.0.

Table 4 shows evaluation results with respect to the attention map extraction layers, evaluated using 10k samples. We select the last self-attention layers of each resolution from the encoder and decoder, and also include the bottleneck layer that divides the encoder and decoder. Regardless of the extraction layer, performance consistently improves over the baseline, while utilizing the self-attention of the final layer yields the best FID and IS results.

C.3. Qualitative results

In addition to the samples in the main paper, we present random samples with SAG from ADM pre-trained with ImageNet 128×128 (Fig. 8), LSUN Cats (Fig. 9), and LSUN Horse (Fig. 10).

Which row do you think shows the better image quality? 1) The top row 2) The bottom row

Figure 4: **An example of a question.** The participants are not told which row is sampled with our method.

D. Human Evaluation Protocol

For the human evaluation of SAG with samples from Stable Diffusion [12], we generate 500 pairs with the empty prompt with or without SAG, and the SAG scale is 1.0 for the samples with SAG. Each pair shares the same seed to make it comparable. We show 50 participants 2 groups of 4 samples, one with SAG and the other without SAG, and ask the participants to select a group having higher image quality. An example of a question is in Fig. 4. Neither the pairs are cherry-picked nor filtered. We also do not perform any post-processing with the responses.

E. Limitations & Future Work

While the increased self-conditioning typically yields results that are more visually appealing to humans, it is important to consider the perspective that the generated images may lack diversity and novelty, a topic that requires discussion. However, at the present stage, the impact of SAG can be effectively moderated by controlling its guidance scale, leading to beneficial applications. Additionally, it requires twice as many feedforward steps, a challenge that is common to CFG [7] and necessitates addressing. A possible solution might involve distilling guidance into diffusion models [8]. This could potentially lessen the computational cost associated with both SAG and CFG, without sacrificing quality.

Moreover, self-attention-based guidance may be more suitable for discrete diffusion models [14, 4], which directly model token probabilities instead of approximating them with continuous values. The integration of these models with our method presents an intriguing topic for future research.

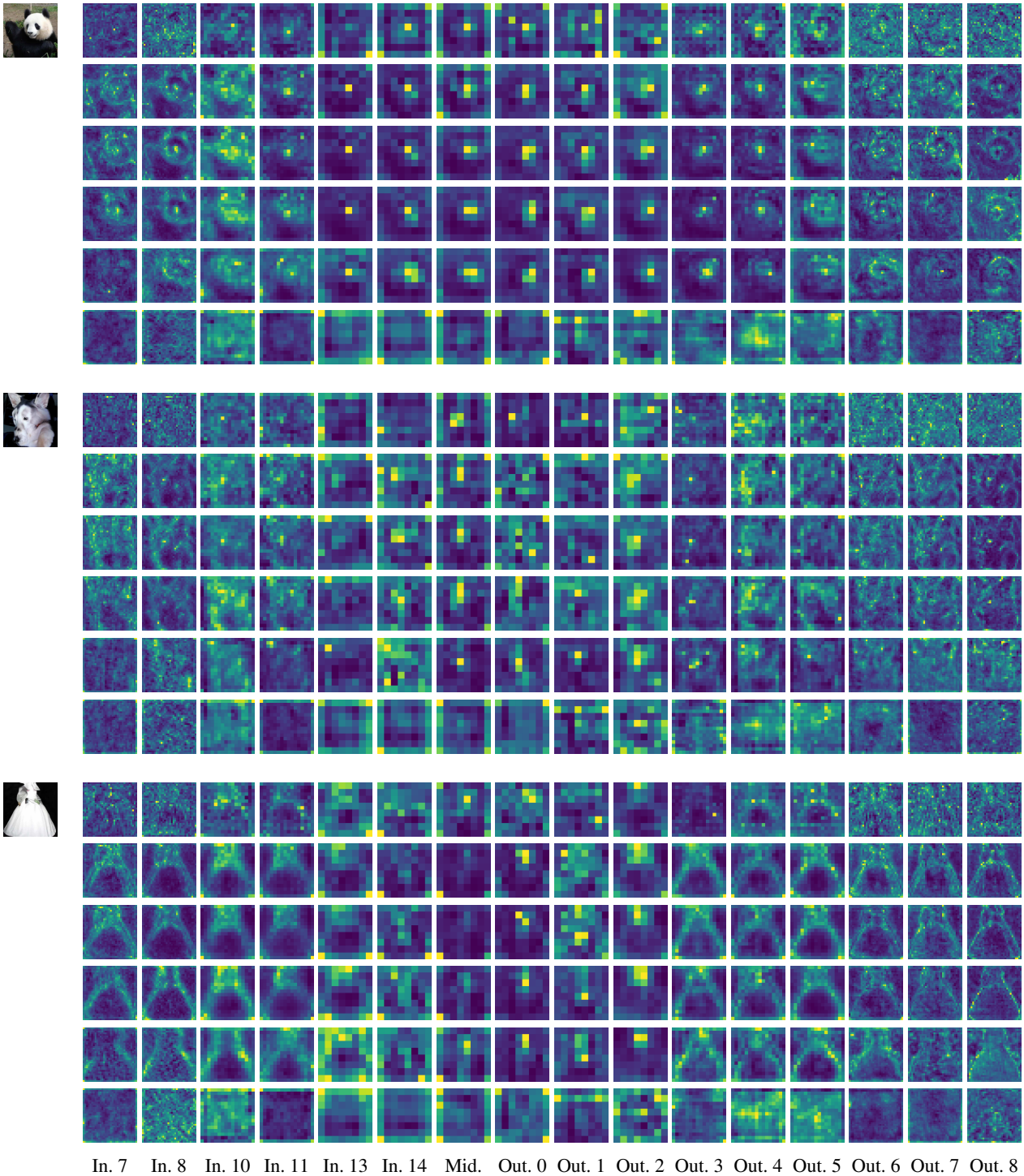


Figure 5: **Attention maps at all the self-attention layers of ADM [3].** In. n, Mid., and Out. n denote the attention map of the nth block of the input blocks, the middle block, and the nth block of the output blocks, respectively.

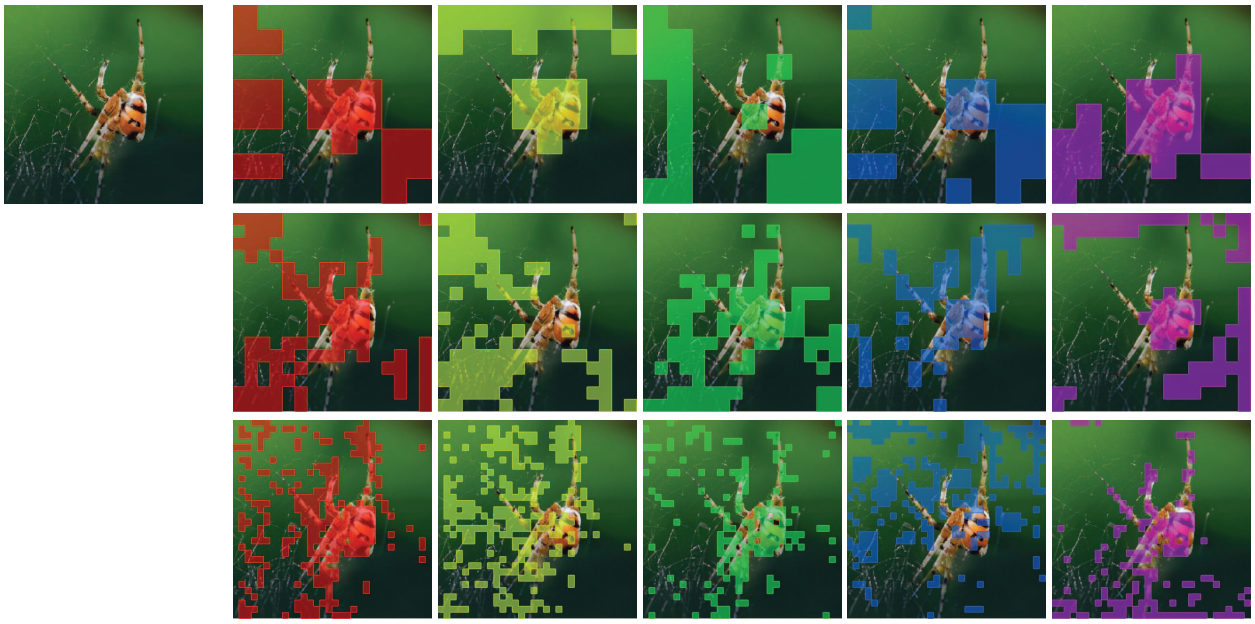


Figure 6: **Visualization of self-attention masks from different layers and heads**. Each row, top to bottom, corresponds to 8×8, 16×16 and 32×32 self-attention layers, respectively.

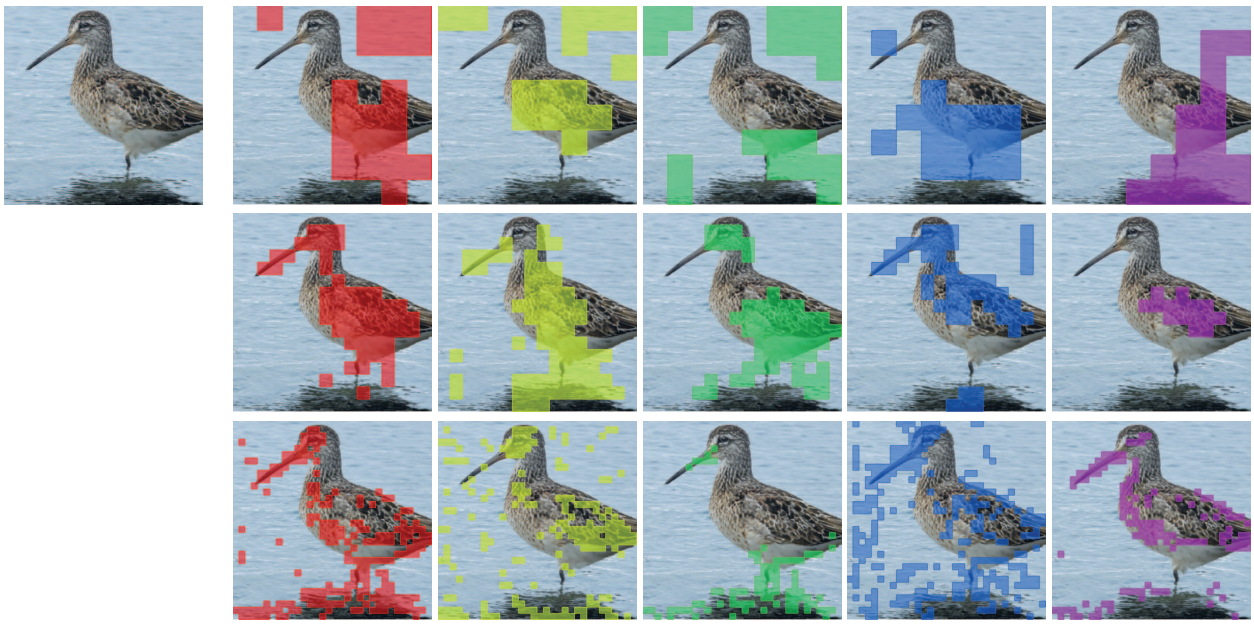


Figure 7: **Visualization of self-attention masks from different layers and heads**. Each row, top to bottom, corresponds to the 8×8, 16×16 and 32×32 self-attention layers, respectively.

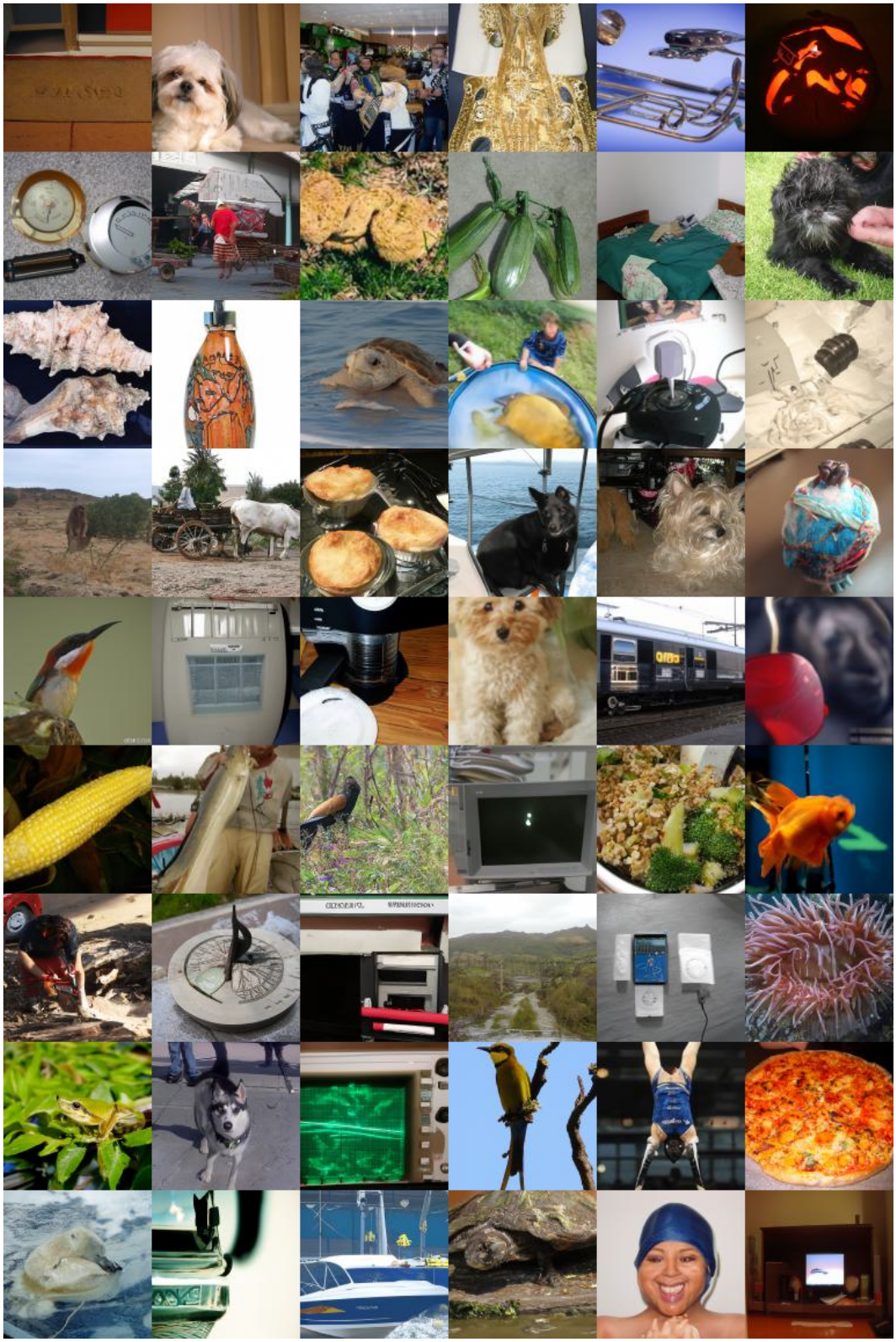


Figure 8: **Uncurated samples with our method.** The results are sampled from ADM [3] conditionally pre-trained in ImageNet [2] 128×128 with self-attention and classifier guidance in combination.

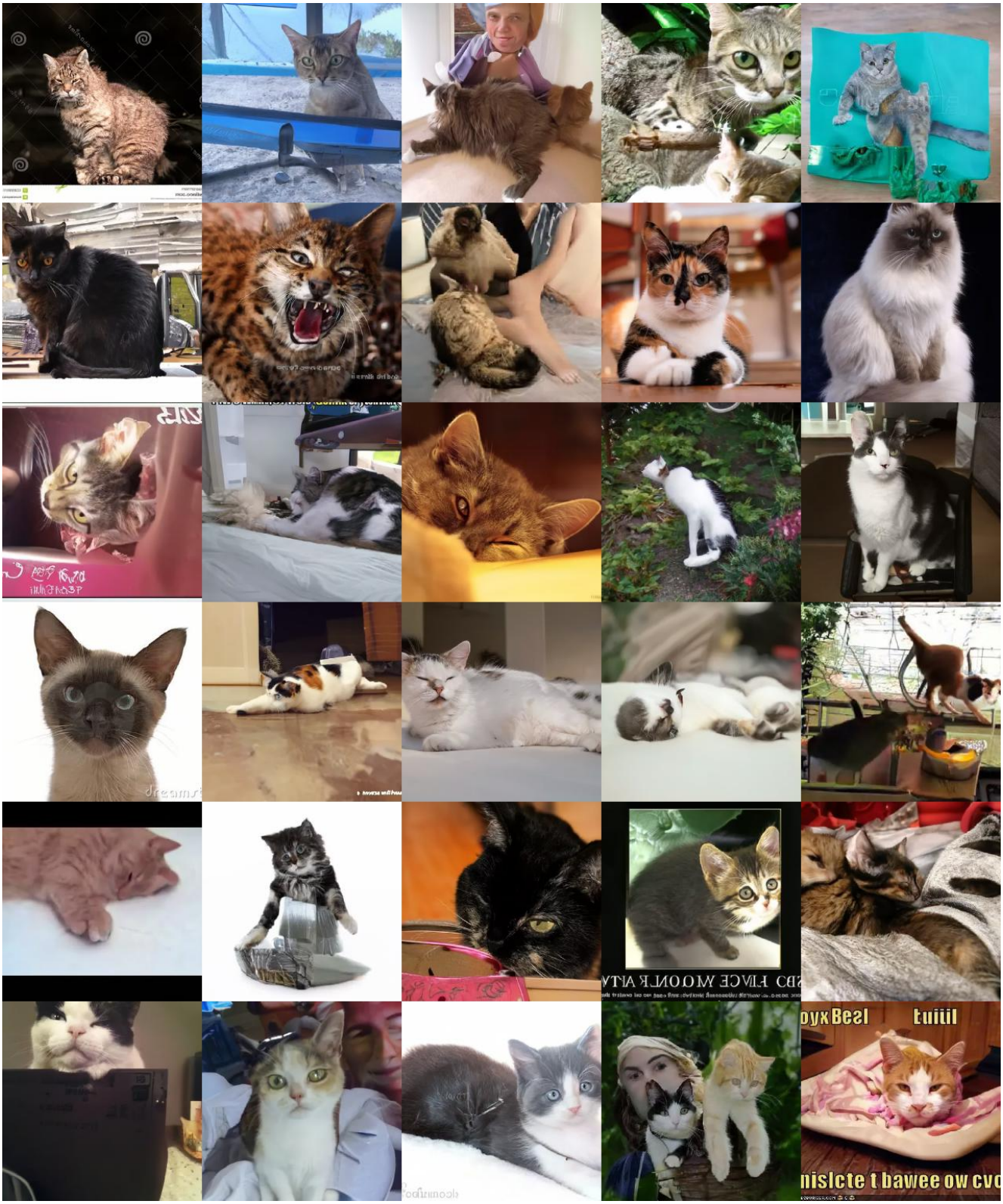


Figure 9: **Uncurated samples with our method.** The results are sampled from ADM [3] pre-trained in LSUN Cat [15] with self-attention guidance.

Figure 10: **Uncurated samples with our method.** The results are sampled from ADM [3] pre-trained in LSUN Horse [15] with self-attention guidance.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 3

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 9

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 1, 2, 3, 4, 6, 9, 10, 11

[4] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, pages 10696–10706, 2022. 5

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 4

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 1, 2

[7] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2, 5

[8] Chenlin Meng, Ruiqi Gao, Diederik P Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022. 5

[9] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pages 8162–8171. PMLR, 2021. 2

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 2

[11] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 2

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2, 5

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[14] Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Improved vector quantized diffusion models. *arXiv preprint arXiv:2205.16007*, 2022. 5

[15] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 10, 11